Genetic Programming Based Learning of Control Rules for Variable Geometry Structures

Koetsu Yamazaki
Dept. of Human & Mechanical Systems Engineering, Kanazawa University, 2-40-20 Kodatsuno, Kanazawa, Ishikawa 920-8667, Japan.
yamazaki@t.kanazawa-u.ac.jp

Sourav Kundu
Dept. of Human & Mechanical Systems Engineering, Kanazawa University, 2-40-20 Kodatsuno, Kanazawa, Ishikawa 920-8667, Japan.
sourav@kenroku.kanazawa-u.ac.jp

Michitomo Hamano
Dept. of Human & Mechanical Systems Engineering, Kanazawa University, 2-40-20 Kodatsuno, Kanazawa, Ishikawa 920-8667, Japan.

ABSTRACT

The difficulties in Variable Geometry (VG) structures are attributable to accuracy requirement(s) in the knowledge of the structural characteristics and coordinated motion of a large number of actuators, among others. The key is to directly control parameters significant to the structure, namely structural forces, strains or relative displacements. The overall control goal, defined by a given objective function to be implemented, requires the most efficient rules to control the key parameters of the structure. These rules are generally obtained by explicit construction. The alternative is to search a large number of candidate rules iteratively or by using an adaptive search procedure. This paper uses the GP paradigm to derive those optimal control rules for VG structures. It describes an application, using the GP for evolution of the key control rules for adaptive structures. The results achieved for a 10-module 2D VG structure are illustrated.

1. Introduction

Variable Geometry (VG) structures have in past provided solutions to many anticipated challenges of space and other applications. A VG structure is a class of adaptive structure that allows controlled alteration of system states to bring about large controlled geometric changes from a random initial geometry or a compact one, to a final distributed geometric configuration. In control of VG structures, the important aspect is observation and control of the basic parameters of the structure that include internal forces, relative deformation, and internal strain energy at a local level. Autonomous distributed control rules seem to provide a framework to achieve this. One begins with a particular rule, which has been constructed to satisfy all the constraints. Then a sequence of "moves" is made in the space of possible rules with each move arranged such that the constraints are still satisfied. In simplest cases, each move is chosen to yield a rule, which is more optimal, or considered having a lower "cost function". Here the structure itself provides the communication between sensors and actuators by its capability to transmit force information. By design, the structure can provide global information by monitoring subsets of internal forces. Some potential challenges of using external forces and inertial acceleration measurements to control VG structures are:

1. Extremely accurate real time knowledge of the structural characteristics is necessary to control the structure. The controller forces are potentially destabilizing and couples with the rigid body motions.
2. High-speed processors are necessary to simultaneously control all actuators and detect erroneous signals or output from faulty sensors and actuators.
3. The inertial actuators are not suitable for quasi-static adjustments and adjustments of internal structural forces or strain energy.
4. The system is not robust.

A control strategy, based on the autonomous distributed criteria is presented here which makes independent adjustments to internal structural forces by actuators that are controlled by *genetically learned* control rules that have hierarchical dependence, to produce global effect from local length adjustments only. It is shown how these rules can be learned by using a GP. Section 2 gives a brief outline VG structures, section 3 describes the GP encoding of the rules followed by a numerical application presented in section 4.

2. Variable Geometry Structure

Figure 1 illustrates a typical example two-dimensional VG structure. The fundamental module of this structure is a rectangular truss composed of an adjustable diagonal member and three fixed members with length L. The VG structure is constructed of n basic modules serially. Thus, the repetition of the basic module in the longitudinal direction forms the whole structure of the VG truss structure. By controlling the lengths of the diagonal member in each module, we can change the configuration of the VG structure into an arbitrary posture in two-dimensional space. The motion the VG structure is highly flexible, but its rigidity is inherently high. The elastic and vibrational properties of the VG truss structures vary depending on its configuration. The initial configuration of the statically determinate structure and its *end-effector* position (point where the force P is applied in Fig. 1), are given by initial length of the adjustable diagonal members. The final optimal configuration of the structure, the final optimal positions of the *end-effector* as well as the optimal trajectory from initial to final, are sought for by minimization of predetermined evaluation criteria. The control rules that the GP evolves take these evaluation criteria into their account. The objective here is to achieve a minimum compliance structure. The GP genetically *breeds* the control rules that interact at local level (in one module of the structure), using local stress and sensitivity information only, while trying to achieve a global behavior, which will adapt to a maximum stiffness posture from any starting arbitrary posture, for any given load direction (angle of force P in Fig. 1). Consequently, maximum stiffness posture will also mean that total strain energy of the structure is minimized. The objective function used for this minimization procedure is :

$$U(\theta_0,\theta_1,\ldots,\theta_{n-1}) = \sum_{m=0}^{n-1} u_m \rightarrow \min \ldots\ldots\ldots\ldots(1)$$

Subject to : ϕ = constant. (refer Fig. 1). Where U = Total strain energy, u_m = strain energy of the m^{th} module, and ϕ = Direction of load. We prescribe the global coordinates (x,y) as situated at the left hand side bottom node of the lowest module. In the simulation procedure of the VG structure we first analyze an initial random configuration of the VG structure by Finite Element Method (FEM). To do this we use the stress values noted by the stress sensors (#0 - #3) (refer Fig. 1). We then determine the control value by control rules. We use these control values as a feedback to re-configure the VG structure. After this, we re-analyze the total structure by the FEM program and test for convergence to given criteria. While convergence criteria are not satisfied, we perform the above steps iteratively as shown in Fig. 4. The control variable is $\theta(m)$ as marked in Fig. 1.

3. Encoding of GP trees

The representation and encoding technique for the autonomous distributed control rules by tree structure of the GP is outlined here. The control program that we propose, consists of three types of tree structures. The hierarchical organization of each GP tree makes it possible to evaluate an output of the actuator, given the present state of the system which pertains to information on key parameters significant to the VG structure. There are four basic trees.

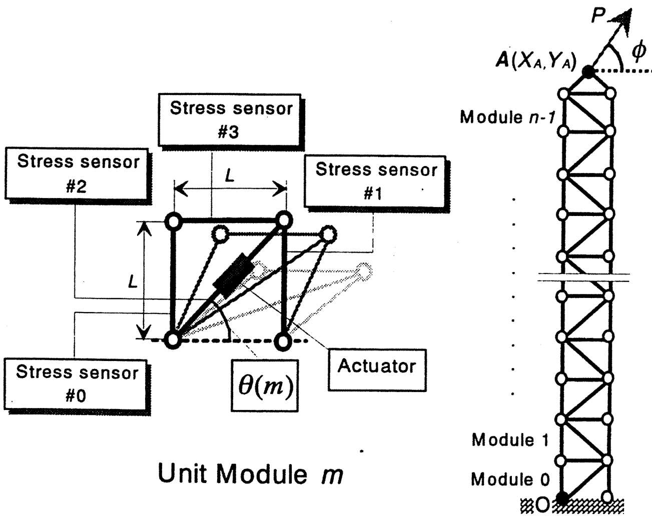


Figure 1 Model of Variable Geometry Structure.

1. Main_tree: This tree is on the top level of the hierarchy and performs Boolean operations on the values returned by trees at the lower level (eg. control_tree condition_tree select_rule_tree). The **Function set** at this level of the Main_tree consists of three elements:

(i) **Prog2**: takes 2 arguments and sequentially executes them.

(ii) **Condition**: takes 2 arguments.

(iii) **Select_rule**: takes one argument.

Terminal set consists of only one operation, which is termed as **Action** that always calls a tree from the lower level of the hierarchy, and returns the value computed by the called tree back to the higher level.

2. Condition_tree and **Select_rule_tree**: As we descend down the hierarchy, the 'action' module in the Main_tree above calls the Condition_tree and/or the Select_rule_tree at the next lower level. These trees store and use the vital information relating to the VG structure of the neighboring modules (upper and lower modules) at the time step when the calculations are being made. The **Function set** of these trees have four of the standard mathematical entities (+, -, *, Not). The **Terminal set** consists of only one expression: *m_stress*. This "*m_stress*", which is shown in equation (2) below is the difference between stress of a member of the module of the VG structure which is presently being considered, and the average stress of corresponding member of the immediate upper and lower modules. This tree returns a value (condition of the module) which is either of 1, 0 or -1 by calculating "*m_stress*" of each member in the corresponding module.

$$m\_stress\_i = \sigma(s,m,i) - \{\sigma(s,m+1,i) + \sigma(s,m-1,i)\}/2 \ldots\ldots\ldots(2)$$

$$s\_stress\_i = \{\sigma(s,m,i) - \sigma(s-1,m,i)\}/\Delta\theta_m \ldots.(i = 0,1,2,3)\ldots\ldots.(3)$$

where $\sigma(s,m,i)$ means stress value of i^{th} member (i=0,1,2,3) of m^{th} Module (current considered Module, m=0,1,..,n-1) at s^{th} adaptive step (present adaptive step, s=0,1,...STEPMAX). The Function set of this tree restricts the control values between -3 and +3, and produces incremental control values for each module of the structure. Note that while calculating "*m_stress*" the VG truss structure is considered with periodic boundaries, that is, bottom-most module becomes an "upper" module for the top-most module.

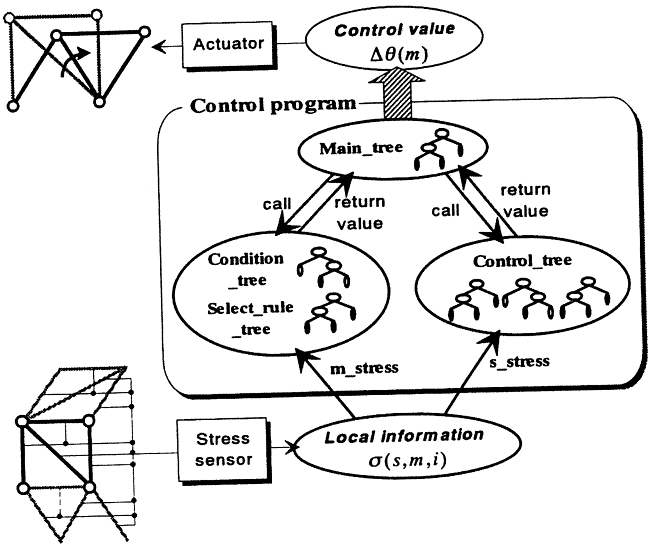


Figure 2 The hierarchical relationship of Main_tree, Condition_tree, Select_rule_tree, Control_tree and others

3. Control_tree : The Control_tree has similar **Function set** as in Condition_tree and Select_rule_tree but the **Terminal set** is the expression: *s_stress*. The "*s_stress*" which is shown in equation (3) above is the stress sensitivity of each member of the module that is calculated from the present iteration step and the previous one. This tree forms the local control rule by calculating *s_stress* of each member of the module. Since this control program can make necessary changes to the control values by considering the different states of the VG structure, as well as the relationship to upper and lower module, a variety of control values can be produced. Fig. 2 shows the hierarchical relationship of the three tree structures. *Condition_tree*, *Select_rule_tree*, *Control_tree* is called by corresponding node of *Main_tree*, and they return their values to *Main_tree*.

New GP crossover operator: In our GP implementation, an improvement of genetic operator was necessary, as convergence was rarely achieved using the standard GP crossover. An individual within a population is composed of four trees, which together implement a trial partial solution of the complete control program, as shown in Fig. 2. Using such structure, it is possible to evolve all the key structures that are necessarily go into making the control program. This is conducive to the formation of "building blocks" concept of John Holland, of useful functionality and enable crossover and other genetic operations, to assemble working implementations of the four operations that go into making the control program that actually control the length changes in the VG structure. Each new individual is created either by copying all six trees of the parent program (10 %) or via crossover between two parent programs (85 %) and mutation (5%). When crossing over, one type of tree is selected at random. The trees of the other types are copied without modification from the first parent to the offspring. The remaining tree is created by crossover between the trees of the chosen type in the standard GP method (Koza, 92) The new tree has the same root as the first parent. Each mating produces a single offspring, most of whose genetic material comes from only one of its parents. Crossover is limited to a single tree at a time in the expectation that this will reduce the extent to which it disrupts the "building blocks" of useful code. Crossing trees with similar functionality trees of another parent is similar to the crossover operator used by John Koza in most of his experiments involving Automatically Defined Functions (ADFs).

4. Numerical Application

In this section we present a numerical application, more complex than that presented in Yamazaki *et al.* (1997), to illustrate and working and effectiveness of the method. We simulate a 10-module VG structure, with dimension of each member given as follows. Length of member of module L= 1000mm, sectional area of member A = 100 mm^2, young's modulus E =205.8GPa. Load condition is assumed as P = 294N, load direction $\phi = 45^\circ$. Table 1 shows the main parameters for the GP simulation. Convergence conditions for the VG truss simulation are as follows.

1. When control value of all modules is 0.
2. When Step of adaptive simulation achieve STEPMAX.

Convergence conditions for the GP is as follows:

1. When best individual doesn't change for 10 successive generations.
2. When number of generations reach GENMAX.

Table 1 Parameters of GP and Simulation.

GP	
Population size	1000
Max generation	50
Selection method	TOURNAMENT
Crossover probability at function node	70%
Crossover probability at any node	15%
Reproduction probability	10%
Mutation probability	5%
Number of training	3
Number of TEST postures	3
Truss simulation	
Max step	100
Unit control value (degree)	0.5

Three types posture shown in Figure 3 and these are used as TEST posture. In calculating fitness, the total strain energy of converged posture is used as standard fitness. As an extension to the Yamazaki *et al.* (1997) paper we now simulate a model with *prescribed load direction* as well as *prescribed load point*. Previously the control program never considered the constraints of load point (*end effector* position). From a study of the VG structure it is evident that the load point constraints will be easily violated while VG structure is being controlled. Therefore we add the constraint that the *end effector* can now return to given load point by modifying control values for all modules in the truss simulation process. Figure 4 shows how this new control program is taken into consideration for the flowchart of this simulation. The shaded box in Fig. 4 is the difference with this implementation from that of Yamazaki *et al.* (1997). The objective function formulation for this example is as follows:

Objective function: $$U = \sum_{m=0}^{n-1} u_m \rightarrow \min$$

Subject to:

$$X_A = L\sum_{m=0}^{n-1}\left\{(-1)^m \cos(2\theta(m))\right\} + L/2 = \text{const.}$$

$$Y_A = L\sum_{m=0}^{n-1}\sin(2\theta(m)) + L/2 = \text{const.}$$

Where U = Total strain energy, and u_m = Strain energy of module m. (XA, YA) = x-y coordinate of *end effector* (load point). Figure 5 shows the initial and final converged postures achieved by using the evolved control rules.

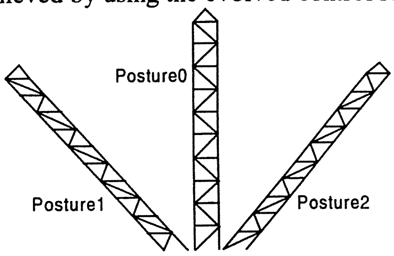


Figure 3 Test postures.

5. Conclusions

In this paper the control rules evolved by the GP for a 10 module VG structure model is shown to work well for prescribed load direction $\phi = 45^\circ$, as well as for prescribed load points. Validity of the control program is confirmed by simulating several initial postures. It is noticed that the total energy of initial posture is steadily decreasing and the VG structure achieves converged posture in an average of 50 steps. In this example we also note that from 30 to 100th step, fluctuations of energy is caused by vibration of some modules. Reducing these vibrations is a direction for future research.

Bibliography

Miura, K. and H, Furuya. 1985. An Adaptive Structure Concept for Future Space Applications. *IAF-85-211. 36th Cong., Internatl. Astronautical Fed., Stockholm,* also *AIAA Journal,* (8)995-1002.

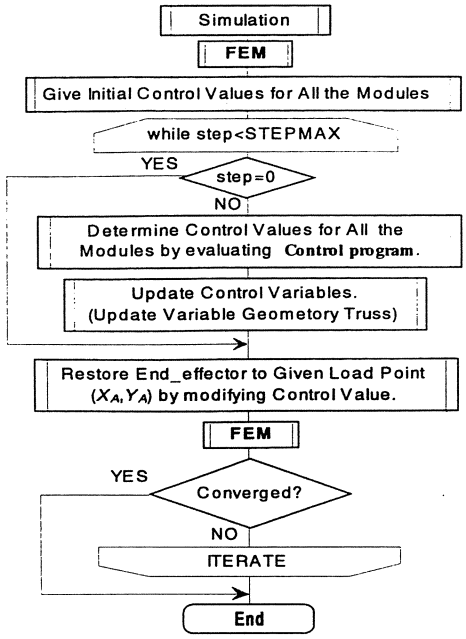
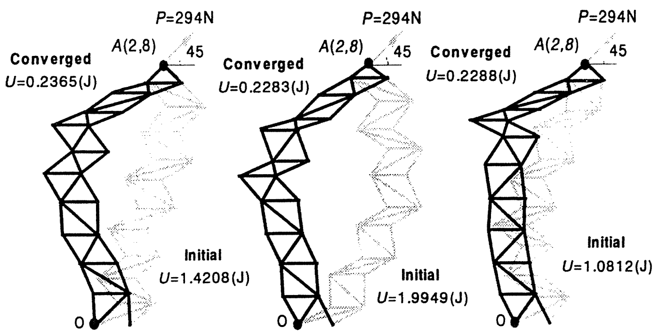


Figure 4 Flowchart for evaluating individuals.

Murotsu, Y., K. Senda and K.,Hisaji. 1990. Optimal Configuration Control of Intelligent Truss Structure, *Proceedings of the First U.S. /Japan Conference on Adaptive Structures, Maui, Hawaii, November 13-15, 1990,* Technomic Publishing Co. Pages 157-175.

Koza, John R. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: The MIT Press.

Yamazaki, K.; Kundu, S. and Hamano, M. 1997. Genetic Learning of Optimal Rules for Posture Adaptation in Variable Geometry Structures. In Hernandez, S., Brebia, C.A. (editors). *Computer Aided Optimum Design of Structures V.* Boston, MA. Computational Mechanics Publications. Pages 239-248.

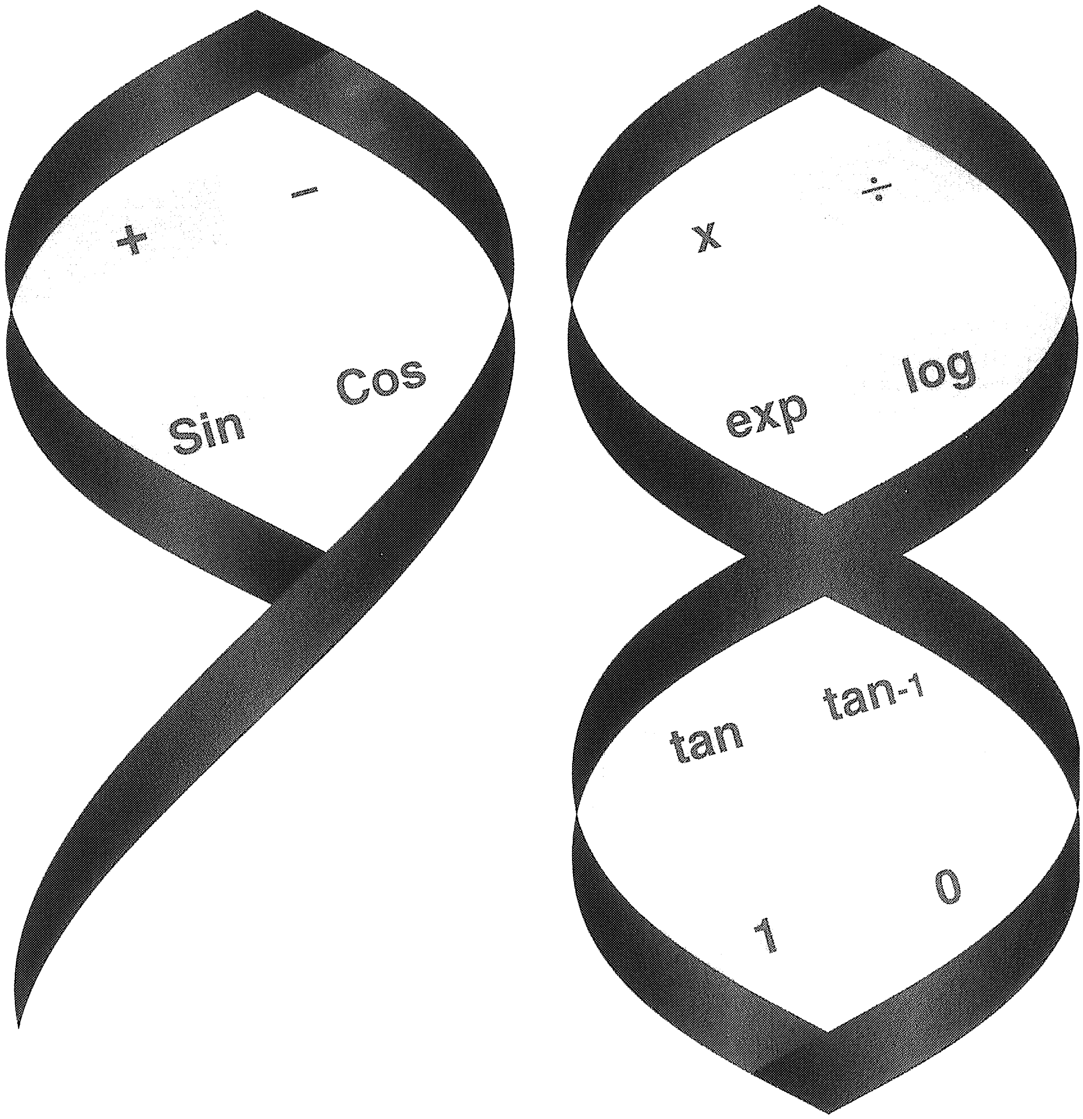


(a)Posture 0 **(b)Posture 1** **(c)Posture 2**

Figure 5 Initial and converged postures, using GP evolved rules.

CONFERENCE PROCEEDINGS

Genetic Programming



edited by
John R. Koza
Wolfgang Banzhaf
Kumar Chellapilla
Kalyanmoy Deb
Marco Dorigo
David B. Fogel
Max H. Garzon
David E. Goldberg
Hitoshi Iba
Rick L. Riolo

Proceedings of the Third Annual Genetic Programming Conference

July 22–25, 1998
University of Wisconsin, Madison, Wisconsin